REMARKS

Claims 1-8 are pending in the present application. No claims have been canceled, amended or added by this response. Reconsideration of the claims is respectfully requested.

The Applicant has submitted a proposed correction to the drawing labeled FIGURE 1. The words "Prior Art" have been added to FIGURE 1. This change will be incorporated into a formal set of drawings, if the Examiner approves the proposed change.

I. Objection to Abstract of the Disclosure

The Examiner has objected to the Abstract of the Disclosure. According to the Examiner, the Title of the Invention should not be on the same page as the Abstract. By this response, the Title of the Invention has been deleted from the Abstract. The Applicant respectfully requests the Examiner to withdraw this objection.

II. 35 U.S.C. § 103, Obviousness

The Examiner has rejected Claims 1-8, under 35 U.S.C. § 103(a), as being unpatentable over U.S. Patent No. 5,418,940 to Mohan ("*Mohan*"). This rejection is respectfully traversed.

As for Claim 1, the Office Action states:

> As per claim 1, Mohan discloses "a method for logging updates to a plurality of data records into discrete pages in nonvolatile storage" (see col. 1, lines 49-52), "wherein a page partially full of data is known as a partial page" as a means for recreating a page from said upon detection of either a partial sector write or a partial page write, (see col. 5, lines 19-21), said method comprising the steps of:
>
> "establishing identical partial pages I and I+1 at the earliest opportunity" as a means for writing of page in first to the last sequence, (see col. 3. lines 43-45), and
>
> "creating identical partial pages I+1 and I+2 with a single, second write operation of D2 to both pages, whereby pages I+1 and I+2 become the new pages I and I+1 for the next logging operation" as the state of the page

can be related to the log records written for that page, in which the LSN of the log record identifying the most recent update to the page is stored in the page itself, (see col. 3, lines 61-65). Mohan does not explicitly disclose "in response to a data segment D larger than the remaining space of a most recent updated partial page I," "partitioning D into a first segment D1 sufficient to fill the remaining space of page I and a second data segment D2, filling page I with a first write operation of its present contents concatenated with D1," "filling page I with a first write operation of its present contents concatenated with D1." On the other hand, Mohan discloses "in response to a data segment D larger than the remaining space of a most recent updated partial page I" as the page had been partially written as where the page LSN sector extent had not been written to DASD, then the page LSN would be less than the LSN of the particular log record for that page, (see col. 4, lines 26-30), "partitioning D into a first segment D1 sufficient to fill the remaining space of page I and a second data segment D2, filling page I with a first write operation of its present contents concatenated with D1" as a means for partitioning a page in the buffer into N sector such that each extent includes a status bit while the last extent also includes an N bit mirror byte of the status bits of all of N extents, (see col. 4, line 66 to col. 5, line 3), and "filling page I with a first write operation of its present contents concatenated with D1" as a means for recreating a page from said log upon detection of either a partial sector write or a partial page, (see col. 5, lines 16-18). It would have been obvious to a person of ordinary skill in the art at the time the invention was made to modify the teachings of Mohan with "in response to a data segment D larger than the remaining space of a most recent updated partial page I," "partitioning D into a first segment D1 sufficient to fill the remaining space of page I and a second data segment D2, filling page I with a first write operation of its present contents concatenated with D1," "filling page I with a first write operation of its present contents concatenated with D1." Such modification would allow the teachings of Mohan to improve the accuracy and the reliability of the data logging method, apparatus, system and computer program, and to provide the list of the LSN of the latest log record written by each transaction, (see col. 7, lines 52-54).

Office Action dated December 31, 2003, pages 3-5.

All limitations of the claimed invention must be considered when determining patentability. *In re Lowry*, 32 F.3d 1579, 1582, 32 U.S.P.Q.2d 1031, 1034 (Fed. Cir. 1994). In comparing *Mohan* to the claimed invention to determine obviousness, limitations of the presently claimed invention may not be ignored. The present invention in Claim 1 recites:

> 1. A method for logging updates to a plurality of data records into discrete pages in non-volatile storage, wherein a page partially full of data is known as a partial page, said method comprising the steps of:
>
> establishing identical partial pages I and I+1 at the earliest opportunity,
>
> in response to a data segment D larger than the remaining space of a most recent updated partial page I, partitioning D into a first segment D1 sufficient to fill the remaining space of page I and a second data segment D2,
>
> filing page I with a first write operation of its present contents concatenated with D1, and
>
> creating identical partial pages I+1 and I-2 with a single, second write operation of D2 to both pages, whereby pages I+1 and I+2 become the new pages I and I+1 for the next logging operation.

None of the features of Claim 1 are taught or suggested by *Mohan*.

Specifically, *Mohan* does not teach or suggest establishing identical partial pages I and I+1 at the earliest opportunity, in response to a data segment D larger than the remaining space of a most recent updated partial page I, partitioning D into a first segment D1 sufficient to fill the remaining space of page I and a second data segment D2, filing page I with a first write operation of its present contents concatenated with D1, and creating identical partial pages I+1 and I-2 with a single, second write operation of D2 to both pages, whereby pages I+1 and I+2 become the new pages I and I+1 for the next logging operation, which are all features of Claim 1. As discussed below, such features are not taught or suggested by *Mohan*. Therefore, Claim 1 is not obvious in view of *Mohan*.

Mohan teaches a method for detecting partial page writes in pages spanning multiple sectors in a storage system. Specifically, *Mohan* discloses a method and means for detecting partial page writes in pages spanning multiple sectors of a sector organized multiple tracked storage facility in a page oriented, log based transaction management

system. This system includes a DASD storage system with pages, and all changes to the pages are written to a log. A page of data may span multiple sectors (e.g., col. 2, lines 44-49 disclose that one page may span 8 sectors).

Mohan teaches that a system can optimize the writing of a data page (e.g., spanning multiple sectors) using an out of order sector writing approach. In other words, in *Mohan*, the system can match the order of writing of the page in a buffer to sectors along an accessed DASD track as it passes under a DASD recording write head. As such, *Mohan* teaches that sectors are not always written in first to last order (e.g., out of order sector writing of a page spanning multiple sectors is disclosed at col. 3, lines 36-54). However, a problem not resolved by *Mohan* is that when sector extents are written out from a buffer to DASD, errors can occur. For example, a full page may not be written (i.e., sectors 1 to 6 may be written, but sectors 7 and 8 may not be written for some reason). Consequently, a significant problem not resolved by *Mohan* is the necessity of being able to detect when a system's intention to write a page of DASD data only results in the writing of a partial page of data into storage (i.e., partial page write error). For example, such a write operation may be to log a change or to update the storage in use. The presently claimed invention resolves that problem.

Mohan described a prior solution (not for a system using out of order sector writing) that used a bit in the first and last bytes of the page. Both bits were initialized to the same value and then inverted before each write operation. Later, when the page was read from the DASD, if the 2-bit values were found to be unequal, then an error had been detected (e.g., see col. 3, lines 6-33). Thus, in that scenario, if the header bit = x, it may be inverted to y when written to storage (sector 1). Sectors 2 to 6 are then written and the last bit of sector 6 is inverted. However, for example, sector 6 is not the end of the page, and inversion of the last bit in sector 6 does not result in the value y. Consequently, when the data is read later, it can be determined that a full page of data was not written (the two bits read will both not equal y). Notably, this approach does not work when out of order sector writing operations are used (e.g., if both the first and last sector are written correctly, but a sector in between is incorrectly written). As such, the approach taught by *Mohan* is directed towards a situation where a status bit is used in each of N sector extents of a page, with each such status bit being set to a common value (e.g., see col. 4,

lines 33-68, col. 5, lines 1-13). In other words, *Mohan* is concerned with the situation where a write operation is not completed properly and only a partial page is written (i.e., an error occurs). For at least this reason, *Mohan* teaches away from the solution provided by the presently claimed invention.

For example, the approach taken in accordance with the presently claimed invention takes into account the likelihood that during a valid logging process, partial pages can result. As such, the present invention is clearly distinguishable from the teachings of *Mohan* because in *Mohan* a write operation has been improperly terminated (i.e., due to an error), but in accordance with the present invention, the last page in the log is typically not full (i.e., there is usually not enough data being logged to fill a complete log page). Typically, the last page in a log is not full of data. If a new logging operation attempts to complete a partial page, it is possible that the only good copy of data in the partial page will be corrupted during the write operation and lost. The presently claimed invention provides solutions for resolving such a problem.

Also, other problems that exist in the prior art are resolved by the present invention. For example, the presently claimed invention strives to minimize the possibility of data corruption during the logging of partial pages, while also minimizing the number of write operations needed. This minimization is accomplished with redundancy, by using the data in the buffer to produce two identical pages partially full of the same data. By using such redundancy, it is possible to ensure that a good copy of the most recent update of the log is always retained. Thus, there is less chance that data will become corrupted during a write operation to the log, which would leave no recent copy of the data for recovery. Therefore, as recited in Claim 1, two identical partial pages I and I+1 are established at the earliest opportunity.

According to the Examiner, *Mohan* teaches this feature at col. 3, lines 43-45. However, *Mohan* does not teach or suggest this feature of Claim 1. *Mohan* actually states that when a page consisting of multiple sectors is being written to DASD, the sectors may be written out of order to optimize performance. There is no mention, teaching or suggestion in *Mohan* that two identical partial pages are being established. Additionally, there is no mention, teaching or suggestion in *Mohan* that two identical partial pages are being established at the earliest opportunity (i.e., purposefully). In

contrast, *Mohan* teaches the writing of complete pages. As such, this approach of *Mohan* is similar to that of the prior solutions in which a partial page results from an error in an operation. In other words, *Mohan* does not teach or suggest that two identical partial pages are established at the earliest opportunity (for a legitimate purpose).

The Examiner also asserts that the claimed feature of "creating partial pages I+1 and I+2 with a single, second write operation of D2 to both pages" is disclosed in *Mohan* at column 3, lines 61-65. Actually, this cited section in *Mohan* only states (continuing to col. 4, line 5) that it is possible to relate the state of a page to a log record written for that page using LSNs. According to this section of *Mohan*, by comparing a page's LSN and the LSN of the logged version, it is possible to determine which version is most up to date. A decision then can be made about whether or not to recover from the log operation, based on such a determination.

In any event, the feature of creating partial pages I+1 and I+2 with a single, second write operation of D2 to both pages, as recited in Claim 1, is not mentioned, taught or suggested by the above-described sections of *Mohan* as cited by the Examiner. Also, as discussed earlier, *Mohan* does not intentionally create partial pages, but is instead concerned with resolving the problem of a partial page occurring in error.

Additionally, the Examiner states that *Mohan* does not explicitly disclose the claimed features of: (1) in response to a data segment D larger than the remaining space of a most recent updated partial page I; (2) partitioning D into a first segment D1 sufficient to fill the remaining space of page I and a second data segment D2; and (3) filling page I with a first write operation of its present contents concatenated with D1. However, the Examiner relates the feature (1) in response to a data segment D larger than the remaining space of a most recent updated partial page I, with teachings in *Mohan* at col. 4, lines 26 to 30.

Actually, *Mohan* only describes (at col. 4, lines 1 through 30) the recovery from a log operation using LSN comparisons, because such solutions assume that a page was written in its entirety (i.e., no partial page write error). Therefore, this section of *Mohan* actually teaches the occurrence of a write error and only a partial page is written as a result. In contrast, the feature in Claim 1 of "in response to a data segment D larger than the remaining space of a most recent updated partial page I, partitioning D into a first

segment D1 sufficient to fill the remaining space of page I and a second data segment D2," is concerned with some of data D being larger than the remaining space of a most recent updated partial page. In accordance with this feature of Claim 1, it is clear that a partial page is not considered the result of an error, but a partial page results in response to a legitimate operation. Thus, in this regard, this step of Claim 1 attempts to fill the remainder of a partial page with a segment D being larger than the remaining space available. This feature is not mentioned, taught or suggested by *Mohan*.

Also, the Examiner relates the feature in Claim 1 of (2) in response to a data segment D larger than the remaining space of a most recent updated partial page I, partitioning D into a first segment D1 sufficient to fill the remaining space of page I and a second data segment D2, with teachings in *Mohan* at column 4, line 66 to col. 5, line 3. Actually, in these sections cited by the Examiner, *Mohan* is not concerned with filling the remaining space of a partial page I with a partitioned data segment D1. In these sections, *Mohan* simply teaches that a page is partitioned into a number of sectors (i.e., because several sectors are needed to store a page). As such, *Mohan* does not teach or suggest in these sections that the system's intention is to fill a partial page (with a portioned data segment), because in *Mohan* partial pages are considered to be the result of write errors.

Additionally, the Examiner relates the feature in Claim 1 of (3) filing page I with a first write operation of its present contents concatenated with D1, with teachings in *Mohan* at col. 5, lines 16-18. Actually, in these sections cited by the Examiner, *Mohan* teaches that all modifications to data stored in the memory locations of a DASD storage system are logged prior to recording the state of changed pages in the memory locations (e.g., concept known as "write ahead logging"). With respect to the claimed feature of filing page I with a first write operation of its present contents concatenated with D1, a partial page had been written previously, and this step in Claim1 overwrites this partial page with its original contents and also enough data to fill the remainder of the partial page. As such, *Mohan* does not teach or suggest in these sections the step of filing page I with a first write operation of its present contents concatenated with D1, as recited in Claim 1.

According to the Examiner, the teachings of *Mohan* (allegedly in the above-cited sections) could have been modified with these features of Claim 1 to allow *Mohan* to

improve the accuracy and the reliability of the data logging method, apparatus, system, and computer program. However, at a minimum, the present invention does not attempt to provide a list of the LSNs of the latest log record written by each transaction. As such, the present invention is not concerned with the problem addressed in *Mohan* in which a write error occurs that results in the writing of a partial page rather than a full page. The present invention provides a solution that assumes that partial pages will occur; not due to a write error, but because data segments are typically not large enough to fill a complete page. As such, the present invention is concerned with the problem of preventing data corruption that may occur in overwriting a partial page with the contents already present in the partial page and also with enough data to fill the remainder of the page. *Mohan* does not deal with or solve such a problem, and as discussed above, does not disclose, suggest or teach the features of establishing identical partial pages I and I+1 at the earliest opportunity, in response to a data segment D larger than the remaining space of a most recent updated partial page I, partitioning D into a first segment D1 sufficient to fill the remaining space of page I and a second data segment D2, filing page I with a first write operation of its present contents concatenated with D1, and creating identical partial pages I+1 and I-2 with a single, second write operation of D2 to both pages, whereby pages I+1 and I+2 become the new pages I and I+1 for the next logging operation, as recited in Claim 1.

The Examiner also rejected independent Claim 4 with essentially the same rationale given above with respect to the rejection of Claim 1. Independent Claim 4 contains features similar to those of Claim 1 and is patentable for the same reasons given above in response to the rejection of Claim 1. Furthermore, Claims 2-3 and 5-8 depend from independent Claims 1 and 4 and are also patentable for the same reasons. Additionally, these claims include other combinations of features not taught or suggested by *Mohan*. Therefore, the Applicant respectfully submits that the rejection of Claims 1-8, under 35 U.S.C. § 103(a), has been overcome and these claims should be allowed.

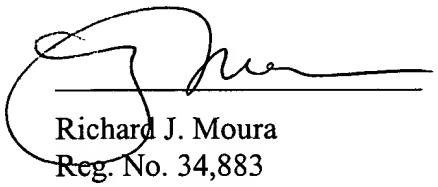
III. Conclusion

It is respectfully urged that the subject application is patentable over *Mohan* and is now in condition for allowance.

The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: March 31, 2004

Respectfully submitted,



Richard J. Moura
Reg. No. 34,883
Carstens, Yee & Cahoon, LLP
P.O. Box 802334
Dallas, TX 75380
(972) 367-2001
Attorney for Applicant